A predictive model for blue bike demand in Metro-Boston Area

Chen Li

Abstract

Blue bike is a public bike sharing system in city of Boston. As of now, the system had deployed more than 400 stations with a fleet of over 4,000 bikes in Boston Metropolitan area. Customer can choose to rent a bike at one of 400 dock stations, and pay via apps from smart phone, as a member for unlimited monthly access, or as casual customer.

In this project, I am particularly interested in predicting blue bike hourly rental using blue bike public available System data. More specially, I will use blue bike trip history data starting from Jan 2019 to Mar 2022. Dataset has been transformed to hourly share demand based on trip starts time.

To make more informative prediction, Hourly blue bike demand data is consolidated with hourly weather observation from Jan 2019 to Mar 2022, provided by national center for environment information, using NOAA climate data record for Boston station WBAN:14739 (Logan airport). The weather information will merge with blue bike demand data by hour of a particular day of the year. Additional information including holidays and weekdays have also been added to the dataset, extracting from federal holiday calendar.

I am interested in understanding how people's riding behaviors have been changed since pandemic time. I will explore questions such as how bike rental volume has being shifted from 2019 to 2022? has people's rental behavior been changed due to Covid -19 quarantine and work from home policy? has location of the most demanding stations been moved though out these three years?

The final hourly blue bike hourly dataset has 28046 records and 14 columns. The goal is to predict blue bike hourly demand using various regression method. Dataset was split randomly as time independent model with 60/40 train/test ratio. 6 algorithms will be applied on dataset: linear regression, Lasso Regression, support vector regression, random forest, XGBoost tree and neural net. The model performance is measured using Root Mean Squared Error (RMSE), R Squared Score (R^2) and lift chart. During the process, we will investigate how model's performance has been changed as we progress to more complex algorithms. We will end up the project with conclusion and future improvement works.

Introduction

The dataset design was inspired by bike sharing demand provided by UCI ML database. Many research and paper have been conducted on this existing dataset. However, there are a few limitations: 1. Dataset is built on capital bike share based in Washington. 2. The data demand was in 2011 & 2012, which is not up to date with current bike demand since Covid-19. Since I moved to Boston in 2020 and has also been an active user of blue bike, I am interested in studying the Boston bike user behaviors by building a new dataset predicting bike hourly demand.

1. Dataset

1.1 Blue Bike Hourly demand data

In this project, I use public available Blue bike system Data that contains all the trip information since 2015 including start & end time, start and end station ID and start and end location in latitude and longitude, Bike ID and user time, as follow:

Example of blue bike

Blue Bikes Comprehensive Trip Histories

We publish downloadable files of Bluebikes trip data each quarter. The data include:

- Trip Duration (seconds)
- Start Time and Date
- Stop Time and Date
- Start Station Name & ID
- End Station Name & ID
- Bike ID
- User Type (Casual = Single Trip or Day Pass user; Member = Annual or Monthly Member)
- Birth Year
- Gender, self-reported by member (Zero=unknown; 1=male; 2=female)

Data dictionary of blue bike system data

To transform the data into hourly demand, I have made following transformation steps:

1. Merged all month data downloaded from Jan 2019 to Mar 2022
2. Extract Year, month, day and hour from start time
3. Group the data by Year, month, day and hour
4. Count the total number of trips in specific Year, month, day and hour to get total hourly demand

1.2 Boston Weather Data from NOAA

national center for environment information provide publicly available hourly weather data in multiple cities across US for research purpose. I used NOAA climate data record for Boston station WBAN:14739 (Logan airport) from Jan 2019 to Mar 2020. The original dataset has 124 features related to Boston hourly weather, I extracted only following hourly related features: exact observation time, hourly dry blub temperature, hourly precipitation, hourly relative humidity, hourly visibility, hourly wind speed, detected by station FM-15 or SY-MT, referring to NOAA indexed data table .

1.3 Final Dataset for Prediction

The final dataset is Boston blue bike hourly demand with weather & Holiday information. The weather information will merge with blue bike demand data by hour of particular day of the year. Additional information including holidays and week days have been added to the dataset, extracting from federal holiday calendar.

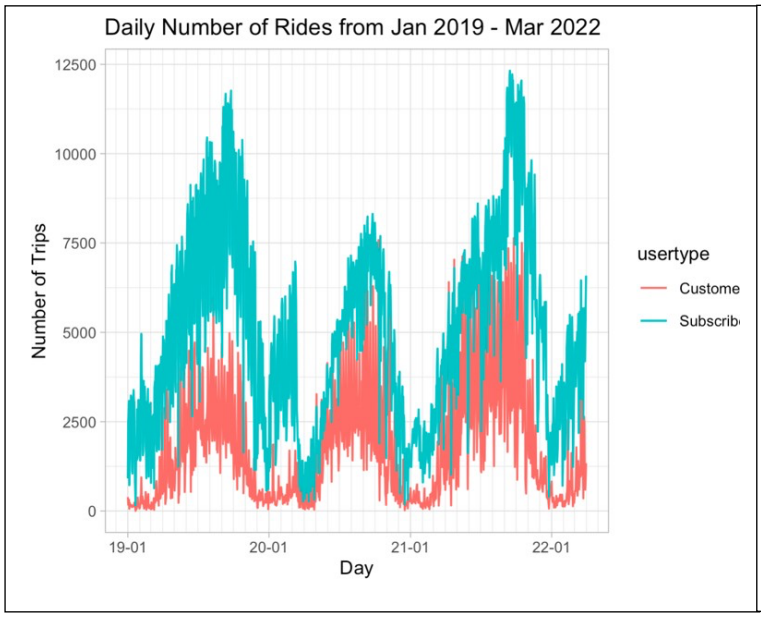
The dataset has 28,0946 rows of bike hourly demand from Jan 2019 to Mar 2022, and 12 columns (11 of them are features, number of ride are the target we are trying to predict. Here is screenshot of final dataset used for prediction

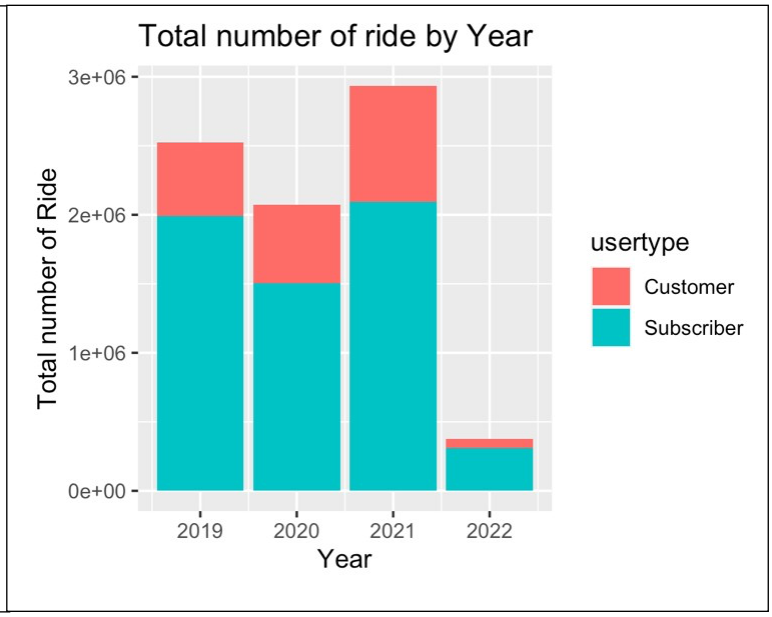
```
Rows: 28,046
Columns: 12
$ Year                     <fct> 2019, 2019, 2019, 2019, 2019, 2019, 2019, 2019, 2019, 2019, 2019, 2019, 201…
$ Month                    <fct> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, …
$ Day                      <fct> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, …
$ Hour                     <fct> 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 2…
$ weekdays                 <fct> Tuesday, Tuesday, Tuesday, Tuesday, Tuesday, Tuesday, Tuesday, Tuesday, Tue…
$ Holiday                  <int> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, …
$ HourlyRelativeHumidity   <int> 93, 89, 93, 89, 97, 93, 83, 80, 75, 67, 60, 51, 47, 47, 44, 48, 52, 53, 55,…
$ HourlyVisibility         <dbl> 5, 5, 7, 4, 3, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, 10, …
$ HourlyDryBulbTemperature <int> 40, 42, 43, 46, 47, 50, 54, 54, 56, 58, 58, 56, 55, 54, 51, 48, 45, 44, 42,…
$ HourlyWindSpeed          <int> 13, 9, 11, 7, 6, 10, 18, 18, 15, 22, 22, 25, 22, 22, 29, 23, 15, 14, 14, 15…
$ HourlyPrecipitation      <dbl> 0.14, 0.09, 0.12, 0.00, 0.03, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.0…
$ num_rides                <int> 7, 24, 5, 3, 1, 2, 7, 10, 29, 42, 79, 94, 130, 181, 147, 160, 103, 93, 47, …
```

2. Data Visualization

Before start to training the model for prediction, I first explored the dataset using plotting and visualization tools to analyze how user patterns has been changed though time.

2.1 Trend of total demand from Jan 2019 to Mar 2022.

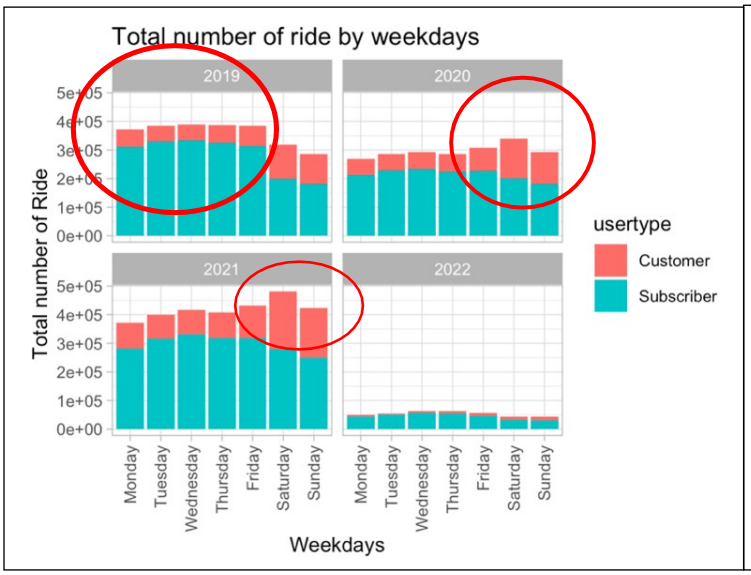


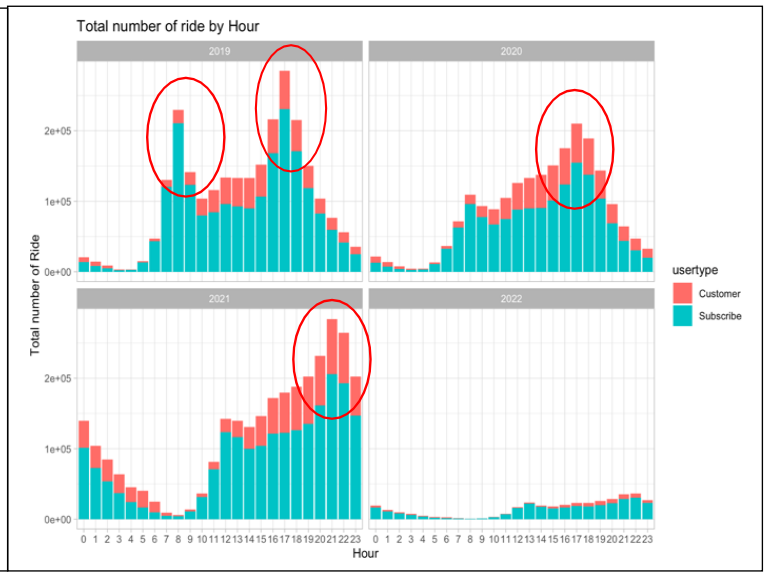


Bike demand has seasonal patterns, it has lowest demand in winter and start to increase to its peak at early fall and decreasing again as approaching to winter. Subscriber has higher demand comparing to non-subscribers (customers) .

On yearly basis, bike demand decreases from 2019 to 2020 and increases significantly in 2021. This is interesting, indicating year might be an important predictor in model

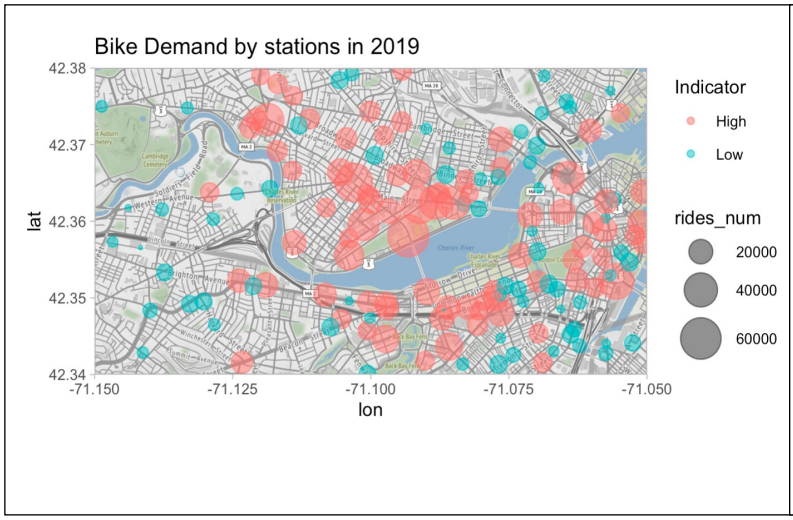
2.2 How Covid - 19 has changed people's living pattern?

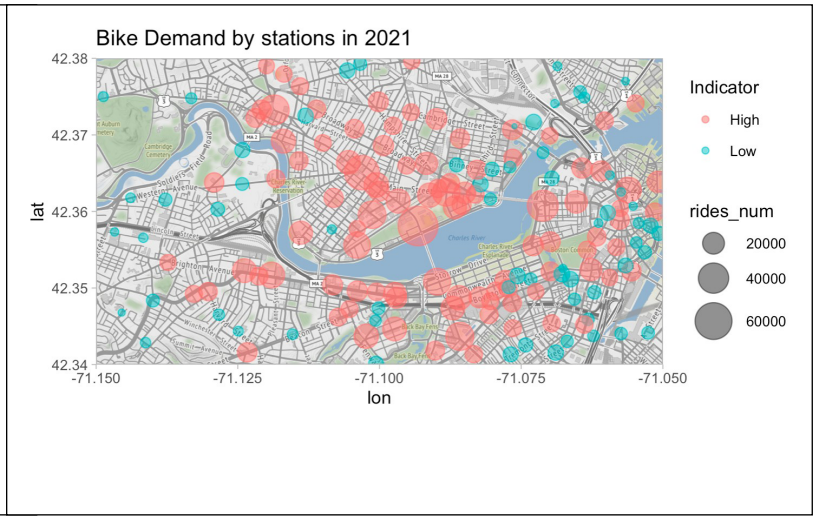




Comparing year 2019 with 2020 & 2021, we can clearly notice a very interesting change. In 2019, the highest hourly demand is usually in weekdays, morning and afternoon peak hours. Since 2020, highest peak of bike demand has gradually shifted to weekends and later afternoon hours, indicating how stay home policy has changed people's living style.

2.3 How bike demand varies at different start station?





The plot is based on hourly bike demand grouped by start station at 2019 vs 2022. In 2019, we can observe most of high demand stations are based in Cambridge and downtown Boston area. While in 2021, there has been less high demand start stations in downtown Boston and more high demand station in Cambridge area near MIT.

3. Data Preprocessing

3.1 Data type

Features (X): Among feature variables, we keep Year, Month, Day, Hour as categorical. Other features including Hourly temperature, precipitation, humidity, visibility & wind speed remains numerical.

Target (Y): number of rides (num_rides) is what we are trying to predict. We keep it as numerical, it has Mean (μ) = 278.9 and standard deviation (σ) = 299.74

3.2 Select variables

Among all features, day variable (value from 1 to 31) is not selected, since month and weekdays carry the same information. Rest of features are kept.

3.3 Dealing with missing values

Due to very small percentage of missing values in dataset (1%), rows with missing values are discarded.

3.4 Train / Test Split

Since the model is predicting future event, I have tried to build a time dependent model by splitting the dataset by time. Train dataset is built on bike demand in year 2019 – 2020 and test dataset is built on bike demand in year 2021. However, the model is performing poorly with out of sample R^2 as 0.02. It might be caused by data in 2019 - 2020 is not well reflects the demand trend in 2021 and we also lost year features in the model.

Hence, instead a **time independent model** is used to build train & test dataset, where each record is split randomly in train & test dataset with 60/40 ratio. This model is performing much better and stable comparing to time dependent model.

Deploy Machine Learning Model for Prediction

1. Linear Regression

As it is a regression problem, I first tried the most basic linear regression with features coefficient been selected based its p value with a threshold of 0.05. Coefficient is approximated using following equation :

$$\hat{\beta} = (X^{\#}X)^{\$\%}X^{\#}Y$$

All features in the model are significant, there are a few categories in month and hour are not significant, however, since they belong to same feature as month and hour, I decided to keep all the features.

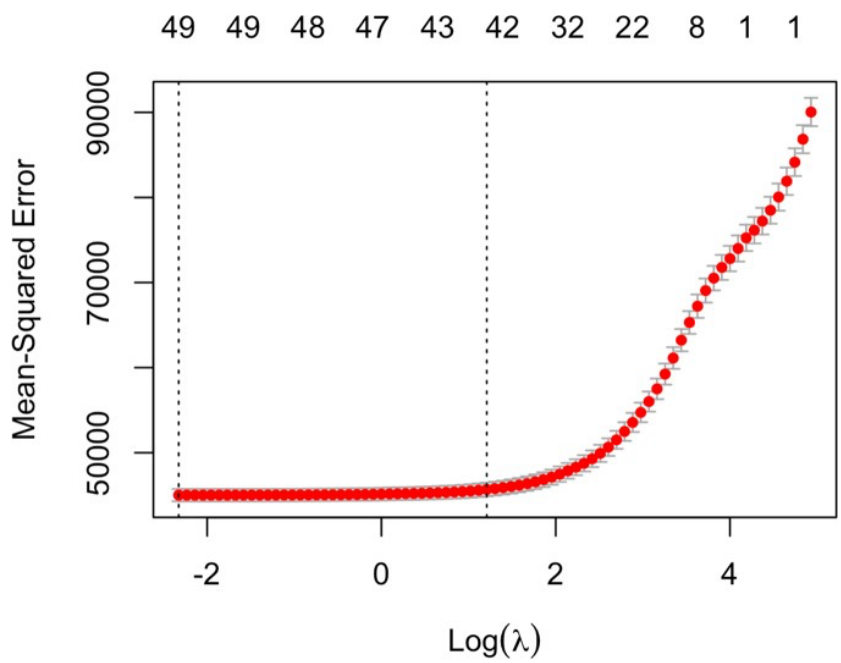
The linear regression model has $\boldsymbol{R^2 = 0.5042}$ and $\boldsymbol{RMSE = 210.62}$

2. Linear Regression with Regulation

Three regression regulation method has been applied on the dataset including Ridge Regression, Lasso and elastic net. Lasso regression has highest R square among these three models. Hence, I decided to use L1 regression method. With N (=28,0946) rows and p (=10) features, we need to optimize coefficient □ that minimize following values, where 11 is shrinkage rate, with its value goes from 0 to oo.

$$\hat{\beta} = argmin \sum_{i=1}^{N} (Y_i - \sum_{j=1}^{p} X_{ij}\beta_j)^2 + \lambda \sum_{j=1}^{p} |\beta_j|$$

To find the best value of 11, I used a 5 fold cross validation method to train 11 from 0 to 1000 and find the best 11 that has the smallest cross validation error in training dataset. The value of 11 that has the smallest cross validation error RMSE is **0.0976**.



The lasso model with 11 = **0.0976** has $R^2 = 0.5042$ and $RMSE = 210.63$

Is Linear regression performing well? Both regression models have low R square around 0.5. Considering the average hourly bike demand is 279, I think RMSE around 210 is considering too high for a productive model. The performance of model shall be further improved by trying various of other algorithms.

3. Support Vector Regression

I picked a more complex algorithm SVR, which is mainly used to compare its performance with linear models and ensembled trees. I choose radial function $f(x) = e^{\$\|-\$-!\|^{"}}$ for kernel function in SVR and kept other model parameters as default in R SVM() function. The SVR has moderately boosted the R square and reduced RMSE, comparing to linear regression.

The SVR model has $\boldsymbol{R^2 = 0.6148}$ and $\boldsymbol{RMSE = 185.66}$

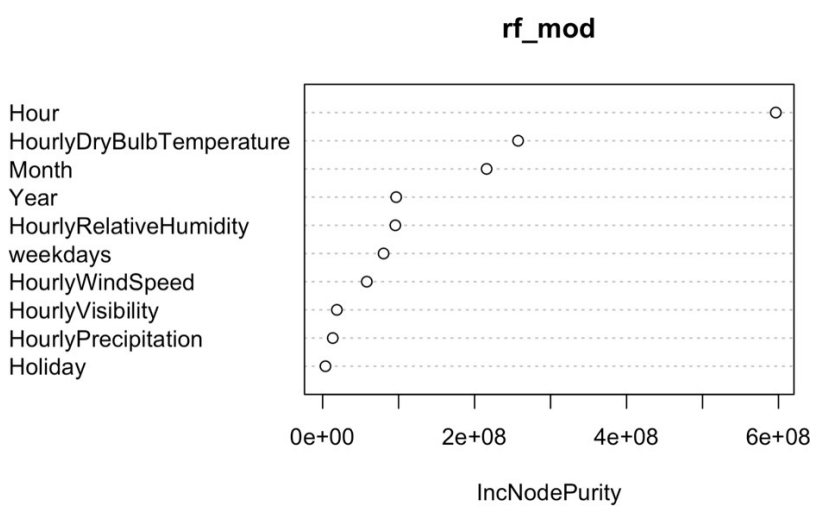
4. Random Forest

Next, I picked Random Forest to compare the performance with previous rigorous regression models and XGB Regression. Random forest is an ensemble method that consist of many decision trees. It is known to significantly reduce the model bias and improve prediction accuracy.

In random forest, I use the default parameters with randomforest() function in R as following: Number of trees to build (ntree) is 500, Number of variables randomly sampled as candidates at each split (mtry) = 3 and Minimum size of terminal nodes (nodesize) = 5.

Random forest takes longest time (>20 mins) among rest of models to build. It has significantly improved the R square and reduced RMSE comparing to SVR and Linear model.

Variance importance plot also gives us an intuition understanding of important variances. We can learn that Hour, temperature and month that the three most important features in prediction



The random forest model has $\boldsymbol{R^2 = 0.8376}$ and $\boldsymbol{RMSE = 120.55}$

5. XGBoost Regression

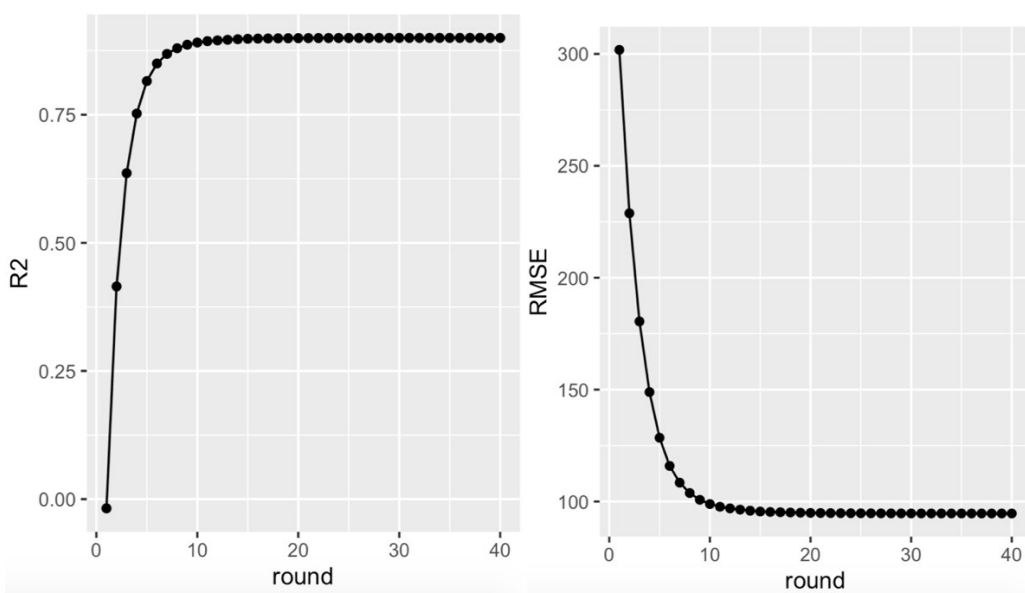
In random forest, we have built n number of trees randomly to reduce bias. In gradient boosted tree, we ensembles a series of tree, where next tree is built to fit the prediction error from previous tree, via gradient descent optimization method.

We used the xgboost package built in R. XGBoost has many tuning parameters. A method of grid search has been applied to find the most optimal tunning parameter. I applied the methodology as following:

I fixed following parameter

- Step size shrinkage ri = 0.3
- Minimal Split loss y = 0
- Maximum depth of a tree = 6
- L2 regularization term on weights 11 = 1
- L1 regularization term on weight a = 0

Then I searched parameter nround (number of boosting round) from 1 to 40, the following plot shows the changes of R square and RMSE as nround increases



Both RMSE and R square starts to stabilize as number of round reaches 20, hence I picked nround = 20 as final parameter.

XGBoost regression improve model performance more from random forest. It also runs quite fast considering we are building a series of regression trees. Here is best tuned model performance.

The XGBoost has $R^2 = 0.8993$ and $RMSE = 94.91$

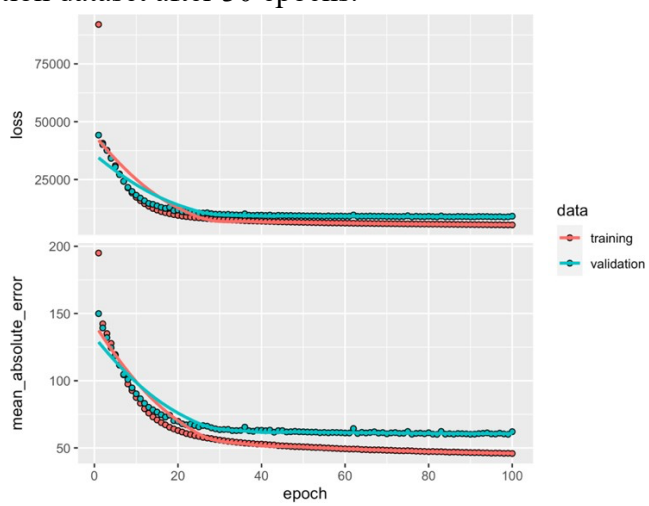
6. Neural Net with Keras

Due to the size of dataset, the build in neural net function in R took more than 1 hour and cannot be completed. Instead, I used Keras API via TensorFlow platform. Keras API provides interactive API to develop deep learning model, including neural net.

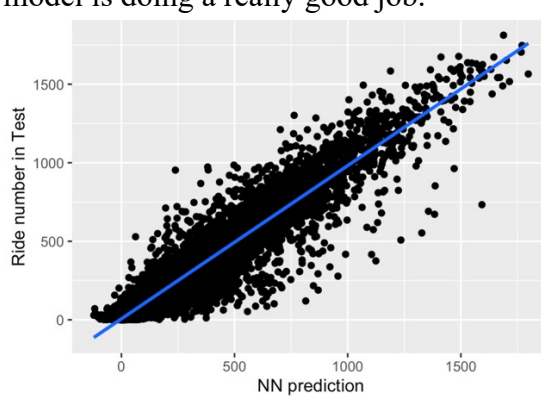
Following steps have been processed to train the model

1. Dummy variables: Categorical variables (Year, month, weekdays, hour) were converted to dummy variables
2. Standardization: all variables are standardized by its mean and standard deviation
3. Model was built with two hidden layer, with 64 neurons in each layer and Relu as activation function
4. Neural Net Model is using mean square error (MSE) as lost function
5. The model is trained for 100 epochs (backward propagation)

The plot below shows the history training process, in each epoch. We can tell the loss function got stabilized on validation dataset after 30 epochs.



Neural net is the best performed model with highest R square and lowest RMSE. When we plot our predicted result from neural net against true value in test dataset, we can clearly see a linear trend indicating neural net model is doing a really good job.



The Neural net model has $\mathbf{R^2 = 0.9019}$ and $\boldsymbol{RMSE = 93.70}$

Conclusion

In this project, we build a new dataset on blue bike hourly demand dataset with boston weather information from scratch. After consolidating the dataset, we first explore the customer behavior using visualization tools. We noticed that customer has seasonal riding behaviors, with highest ride demand in late fall and lowest ride demand in coldest time of winter. We also analyze how average demand has been changed from 2019 to 2022, as well as how customer ride behavior has shifted since pandemic time, during which demand peak goes from morning / afternoon rush hour during weekdays to later afternoon during weekends.

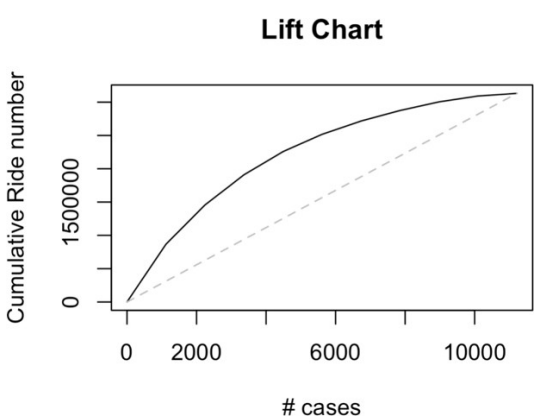
We then preprocess the data by converting categorical and numerical variables, fixing the missing values and build a time independent model by creating a random train/test split.

6 regression models have been built on blue bike demand dataset for predicting hourly demand, including Linear regression, linear regression with regulation (L1 lasso), Support Vector Regression, Random Forest, XGBoost tree and neural net. 5-fold cross validation has also been applied on Lasso regression and grid search has been tested on XGBoost to find out the best tuned parameters. Lasso regression is built with most optional 11 = 0.0976. XGBoost is built with most optimal training around = 20 and max depth = 20 in each ensemble tree. Neural net is built with Keras with Tensflow API, with two hidden layers of 64 neurons in each layer and relu as activation function.

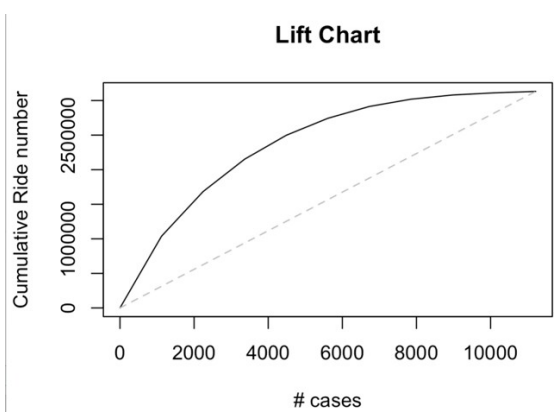
The model performance was measure in three metrics, RMSE, R Square and lift chart. The table below compares the model performance.

Metrics	*Linear Regression*	*LR-Lasso Regression*	*SVR*	*Random Forest*	*XGBoost Tree*	*Neural Net*
RMSE	210.61	210.63	185.66	120.55	94.90	**93.70**
R Square	0.5042	0.5042	0.6148	0.8379	0.8993	**0.9018**

When comparing the lift chart of linear regression vs Neural net, we can clearly see a higher lift from neural net.



Lift Chart – Linear Regression



Lift Chart – Neural net

Among all 6 models, Neural net outperforms rest of models with highest RMSE = 93.7 and R square = 0.9, so it is most recommended model. XGBoost has very close R square = 0.89 and RMSE = 94.9, it performs training much faster than neural net. Hence, both models could be recommended for bike demand prediction.

Though both models have limitations: 1. Neural net and XGBoost tree are known as black box models, they are hard to interpret and understand how features have impacted the performance. 2. Both models take quite amount of time for training to be completed.

Future work

Due to limitation of time, the model was built as a regression issue with selected algorithms. In future, the model could be transferred into a classification problem. It could make supply chain planning efficient by predict high vs low demand for bike stock and supply at various station. The target value "number of ride" can be converted to a binary variable as "high" vs "low, where "high" means hourly demand is higher than total average demand and vice versa.

I would also like to test with a time dependent model. We could use history to predict future demand, for example, use 2020 – 2021 demand data to predict 2022 demand. Due to the nature of this datasets, where the hourly rental demands actually decrease from 2019 to 2020, and increase again in 2021. Training dataset with 2019 – 2020 demand fails to capture the future demand accurately, hence perform very poorly. In the future, it would be worth to try how the model perform by using 2020 – 2021 demand to predict on 2022 demand, after we got complete 2022 demand data.

In the end, Time series method could also be applied on dataset, by predicting hourly bike demand using method such as seasonal trend analysis, linear fit trend and Exponential Smoothing Techniques.

Reference

- UCI ML repository Bike sharing dataset: http://archive.ics.uci.edu/ml/datasets/Bike+Sharing+Dataset
- Blue bike trip history data: https://s3.amazonaws.com/hubway-data/index.html
- Boston Weather provided by nation center for environmental information: https://www.ncdc.noaa.gov/cdo-web/datasets/GHCND/stations/GHCND:USW00014739/detail